

# The RSPLIB Project – From Research to Application

Thomas Dreibholz, Martin Becke  
University of Duisburg-Essen, Institute for Experimental Mathematics  
Ellernstrasse 29, 45326 Essen, Germany  
{dreibh,martin.becke}@iem.uni-due.de

**Abstract**—The goal of the RSPLIB project has been to design and develop an Open Source implementation of Reliable Server Pooling (RSerPool), the new IETF framework for server pool and session management. While its original intention had been the research on RSerPool optimizations and protocol improvements, it has meanwhile become a production-ready platform which is already used by a couple of applications. In our proposed demo for the IEEE GLOBECOM 2010, we would like to introduce RSerPool and show how applications can benefit from using the features provided by RSPLIB.<sup>1</sup>

**Keywords:** Reliable Server Pooling, Service Availability, Session Failover, Load Distribution, Demonstration

## I. SCOPE

Service availability is becoming increasingly important in today's Internet. However, there had been no generic, standardized approaches for managing the availability of Internet-based services. Instead, each developer of an availability-critical application had to continuously re-invent the wheel again and again. To overcome this problem, the IETF RSerPool WG had been founded to develop Reliable Server Pooling (RSerPool, [1]), an application-independent framework for managing server pools and sessions with such pools. RSerPool is strongly related to the Stream Control Transmission Protocol (SCTP, [2]), which is used for almost all RSerPool communications.

## II. THE RSPLIB IMPLEMENTATION

In order to examine ideas for and provide improvements to the upcoming RSerPool framework, the RSPLIB project<sup>2</sup> [3] had been funded at the Institute for Experimental Mathematics of the University of Duisburg-Essen, Germany. Its goal had been to design and develop a GPLv3-licensed Open Source implementation of RSerPool, which currently supports Linux, FreeBSD, MacOS X and Solaris. Meanwhile, RSPLIB has not only become the reference implementation of the IETF RSerPool WG but has also been accepted by Ubuntu Linux to be included in their official installation repositories.

Until today, RSPLIB has been used to research various aspects of RSerPool, e.g. load distribution [4], [5], session failover handling [6], pool management [7], [8], security [9] and applications [10], [11]. This research on RSerPool has led to various refinements of the RSerPool specification, finally

resulting in the standardization of RSerPool as RFCs [1], [12]–[15]. While there is still ongoing research work on RSerPool using RSPLIB, our implementation is nowadays also used for research on the SCTP protocol [2] and its extensions [16]–[18]: RSerPool utilizes a large subset of the advanced transport features provided by SCTP; RSPLIB is therefore a highly useful and representative application for realistic SCTP performance evaluations.

## III. OUR DEMONSTRATION

The RSPLIB package provides a set of demo applications. At the IEEE GLOBECOM 2010, we would like to demonstrate two of them: the Fractal Generator Demo and the Scripting Service.

### A. Fractal Generator Demo

The Fractal Generator Demo is a system to illustratively demonstrate the functionalities of RSerPool applications and protocols as well as of the underlying SCTP protocol. A screenshot of the demo system is depicted in figure 1. The control GUI (left-hand side) allows to turn components on or off and displays the current status of the communication flows among the components. The example application service provides the computation of fractal graphics [19]. This illustrative application makes the effect of introduced component failures – and their handling by RSerPool and the underlying SCTP protocol – easily visible. The demo system includes appropriate dissectors and filters for WIRESHARK [20] to even show packet flows as well as to display the contents of certain packets.

The components of the setup may reside on different physical PCs. For our demo at the IEEE GLOBECOM 2010, we are going to distribute them to two laptop PCs locally as well as some further remote PCs globally, connected over the Internet. That is, instead of just showing an artificial lab demo, we are going to present a real-life Internet application scenario having its components distributed all over the world!

Using the Fractal Generator Demo, we will introduce the RSerPool framework, its protocols and the interaction with its applications to the audience.

### B. Scripting Service

A more advanced application of RSPLIB/RSerPool is the so-called Scripting Service. This service provides the remote

<sup>1</sup>Parts of this work have been funded by the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG).

<sup>2</sup>RSPLIB homepage: <http://tdrwww.iem.uni-due.de/dreibholz/rserpool/>.

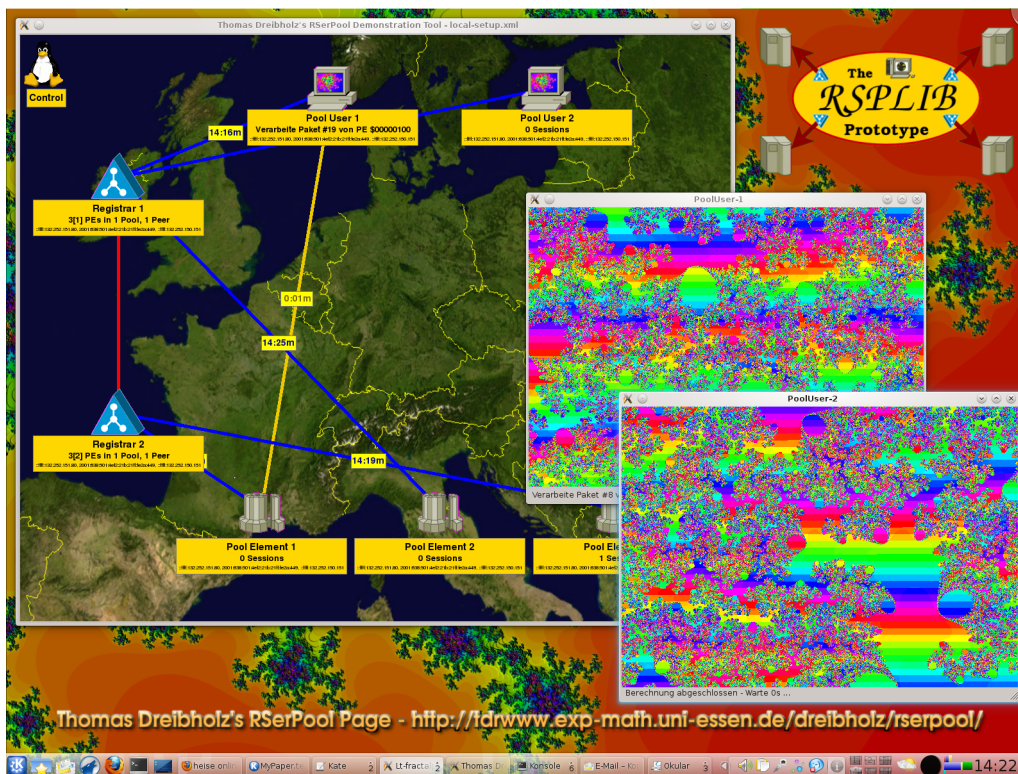


Figure 1. The RSerPool Fractal Generator Demo

computation of user workload, which is e.g. utilized by the SIMPROCTC tool-chain [11] to distribute simulation runs into a compute pool. But unlike highly complex Grid computing solutions, RSPLIB makes the task of setting up such a computation pool very easy and convenient for the user. Particularly, this is performed by the automatic configuration features of RSerPool. That is, beside pure “demo” applications like the fractal graphics computations, RSPLIB can also be applied for real-world tasks (e.g. the RSPLIB-based SIMPROCTC is already in production use at several universities).

In the second part of our demo at the IEEE GLOBE-COM 2010, we would like to show how the Scripting Service can be used to distribute real workload, on the example of processing POV-RAY [21] ray-tracing computations into a globally distributed pool – with respect to utilizing SCTP protocol extensions like CMT-SCTP [16], [18], [22] and CMT/RP-SCTP [17] for improved transport performance. In particular, we would like to demonstrate to the audience how RSPLIB can be applied easily to adapt this service for their *own* applications – with a configuration effort of just a few minutes!

Of course, all software as well as an installation tutorial will be provided as Open Source, in order to provide a long-term value to the audience. Our presentation will also be a demonstration of our achieved project goal to bring our ideas for RSerPool and SCTP from research into application.

#### IV. EQUIPMENT, FACILITY AND SPACE REQUIREMENTS

For our demo, we only require a table to set up two laptop computers and some networking equipment. All equipment is brought by ourselves. Furthermore, electrical outlets and Internet access are necessary.

The following additional items would be nice to have for our demonstration, but are not mandatory:

- Poster board to mount a DIN A0 (84.1×118.9 cm, 33.1×46.8 inches) poster as well as mounting materials,
- Wired Internet access by using Ethernet,
- Global IPv4 addresses (i.e. no NAT) as well as
- IPv6 connectivity.

#### V. CONCLUSION

Reliable Server Pooling (RSerPool) is the new IETF framework for server pool management and session failover handling. In particular, it can be used for realizing highly available services and load distribution. RSPLIB is our implementation of RSerPool. In our demonstration, we would like to give the audience an introduction to RSerPool and RSPLIB, as well as to show how RSPLIB can be applied easily for own applications – with respect to further improving the application performance by utilizing certain features of the underlying SCTP transport.

#### REFERENCES

- [1] P. Lei, L. Ong, M. Tüxen, and T. Dreibholz, “An Overview of Reliable Server Pooling Protocols,” IETF, Informational RFC 5351, Sep. 2008.
- [2] R. Stewart, “Stream Control Transmission Protocol,” IETF, Standards Track RFC 4960, Sep. 2007.

- [3] T. Dreibholz, "Reliable Server Pooling – Evaluation, Optimization and Extension of a Novel IETF Architecture," Ph.D. dissertation, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, Mar. 2007.
- [4] T. Dreibholz and E. P. Rathgeb, "On the Performance of Reliable Server Pooling Systems," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN) 30th Anniversary*, Sydney/Australia, Nov. 2005, pp. 200–208, ISBN 0-7695-2421-4.
- [5] —, "On Improving the Performance of Reliable Server Pooling Systems for Distance-Sensitive Distributed Applications," in *Proceedings of the 15. ITG/GI Fachtagung Kommunikation in Verteilten Systemen (KiVS)*, Bern/Switzerland, Feb. 2007, pp. 39–50, ISBN 978-3-540-69962-0.
- [6] —, "Overview and Evaluation of the Server Redundancy and Session Failover Mechanisms in the Reliable Server Pooling Framework," *International Journal on Advances in Internet Technology (IJAIT)*, vol. 2, no. 1, pp. 1–14, Jun. 2009, ISSN 1942-2652.
- [7] —, "An Evaluation of the Pool Maintenance Overhead in Reliable Server Pooling Systems," *SERSC International Journal on Hybrid Information Technology (IJHIT)*, vol. 1, no. 2, pp. 17–32, Apr. 2008, ISSN 1738-9968.
- [8] X. Zhou, T. Dreibholz, E. P. Rathgeb, and W. Du, "Takeover Suggestion – A Registrar Redundancy Handling Optimization for Reliable Server Pooling Systems," in *Proceedings of the 10th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2009)*, Daegu, South Korea, May 2009, pp. 168–173, ISBN 978-3-642-01202-0.
- [9] T. Dreibholz, X. Zhou, E. P. Rathgeb, and W. Du, "A PlanetLab-Based Performance Analysis of RSerPool Security Mechanisms," in *Proceedings of the 10th IEEE International Conference on Telecommunications (ConTEL)*, Zagreb/Croatia, Jun. 2009, pp. 213–220, ISBN 978-953-184-131-3.
- [10] X. Zhou, T. Dreibholz, M. Becke, J. Pulinthanath, E. P. Rathgeb, and W. Du, "The Software Modeling and Implementation of Reliable Server Pooling and RSPLIB," in *Proceedings of the 8th ACIS Conference on Software Engineering Research, Management and Applications (SERA)*, Montréal, Québec/Canada, May 2010, pp. 129–136, ISBN 978-0-7695-4075-7.
- [11] T. Dreibholz and E. P. Rathgeb, "A Powerful Tool-Chain for Setup, Distributed Processing, Analysis and Debugging of OMNeT++ Simulations," in *Proceedings of the 1st ACM/ICST International Workshop on OMNeT++*, Marseille/France, Mar. 2008, ISBN 978-963-9799-20-2.
- [12] R. Stewart, Q. Xie, M. Stillman, and M. Tüxen, "Aggregate Server Access Protocol (ASAP)," IETF, RFC 5352, Sep. 2008.
- [13] Q. Xie, R. Stewart, M. Stillman, M. Tüxen, and A. Silvertan, "Endpoint Handlespace Redundancy Protocol (ENRP)," IETF, RFC 5353, Sep. 2008.
- [14] T. Dreibholz and M. Tüxen, "Reliable Server Pooling Policies," IETF, RFC 5356, Sep. 2008.
- [15] T. Dreibholz and J. Mulik, "Reliable Server Pooling MIB Module Definition," IETF, RSerPool Working Group, RFC 5525, Apr. 2009.
- [16] T. Dreibholz, M. Becke, E. P. Rathgeb, and M. Tüxen, "On the Use of Concurrent Multipath Transfer over Asymmetric Paths," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Miami, Florida/U.S.A., Dec. 2010.
- [17] T. Dreibholz, M. Becke, J. Pulinthanath, and E. P. Rathgeb, "Applying TCP-Friendly Congestion Control to Concurrent Multipath Transfer," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, Perth/Australia, Apr. 2010, pp. 312–319, ISSN 1550-445X.
- [18] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 951–964, Oct. 2006, ISSN 1063-6692.
- [19] H.-O. Peitgen and P. Richter, *The Beauty of Fractals*, XII ed. Heidelberg/Germany: Springer-Verlag, 1986, ISBN 978-3-540-15851-6.
- [20] Wireshark, "Wireshark: The World's Most Popular Network Protocol Analyzer," 2010.
- [21] POV-Team, "POV-Ray – The Persistence of Vision Ray Tracer," 2010.
- [22] M. Becke, T. Dreibholz, J. Iyengar, P. Natarajan, and M. Tüxen, "Load Sharing for the Stream Control Transmission Protocol (SCTP)," IETF, Network Working Group, Internet-Draft Version 00, Jul. 2010, draft-tuexen-tsvwg-sctp-multipath-00, work in progress.